

032534
032535
032536
032537
032538
032539
032540
032541
032542
032543
032544
032545
032546
032547
032548
032549
032550
032551
032552
032553
032554
032555
032556
032557
032558
032559
032560
032561
032562
032563
032564
032565
032566
032567
032568
032569
032570
032571
032572
032573
032574
032575
032576
032577
032578
032579
032580
032581
032582
032583
032584
032585
032586
032587
032588
032589
032590
032591
032592
032593
032594
032595
032596
032597
032598
032599
032600
032601
032602
032603
032604
032605
032606
032607
032608
032609
032610
032611
032612
032613
032614
032615
032616
032617
032618
032619
032620
032621
032622
032623
032624
032625
032626
032627
032628
032629
032630
032631
032632
032633
032634
032635
032636
032637
032638
032639
032640
032641
032642
032643
032644
032645
032646
032647
032648
032649
032650
032651
032652
032653
032654
032655
032656
032657
032658
032659
032660
032661
032662
032663
032664
032665
032666
032667
032668
032669
032670
032671
032672
032673
032674
032675
032676
032677
032678
032679
032680
032681
032682
032683
032684
032685
032686
032687
032688
032689
032690
032691
032692
032693
032694
032695
032696
032697
032698
032699
032700
032701
032702
032703
032704
032705
032706
032707
032708
032709
032710
032711
032712
032713
032714
032715
032716
032717
032718
032719
032720
032721
032722
032723
032724
032725
032726
032727
032728
032729
032730
032731
032732
032733
032734
032735
032736
032737
032738
032739
032740
032741
032742
032743
032744
032745
032746
032747
032748
032749
032750
032751
032752
032753
032754
032755
032756
032757
032758
032759
032760
032761
032762
032763
032764
032765
032766
032767
032768
032769
032770
032771
032772
032773
032774
032775
032776
032777
032778
032779
032780
032781
032782
032783
032784
032785
032786
032787
032788
032789
032790
032791
032792
032793
032794
032795
032796
032797
032798
032799
032800
032801
032802
032803
032804
032805
032806
032807
032808
032809
032810
032811
032812
032813
032814
032815
032816
032817
032818
032819
032820
032821
032822
032823
032824
032825
032826
032827
032828
032829
032830
032831
032832
032833
032834
032835
032836
032837
032838
032839
032840
032841
032842
032843
032844
032845
032846
032847
032848
032849
032850
032851
032852
032853
032854
032855
032856
032857
032858
032859
032860
032861
032862
032863
032864
032865
032866
032867
032868
032869
032870
032871
032872
032873
032874
032875
032876
032877
032878
032879
032880
032881
032882
032883
032884
032885
032886
032887
032888
032889
032890
032891
032892
032893
032894
032895
032896
032897
032898
032899
032900
032901
032902
032903
032904
032905
032906
032907
032908
032909
032910
032911
032912
032913
032914
032915
032916
032917
032918
032919
032920
032921
032922
032923
032924
032925
032926
032927
032928
032929
032930
032931
032932
032933
032934
032935
032936
032937
032938
032939
032940
032941
032942
032943
032944
032945
032946
032947
032948
032949
032950
032951
032952
032953
032954
032955
032956
032957
032958
032959
032960
032961
032962
032963
032964
032965
032966
032967
032968
032969
032970
032971
032972
032973
032974
032975
032976
032977
032978
032979
032980
032981
032982
032983
032984
032985
032986
032987
032988
032989
032990
032991
032992
032993
032994
032995
032996
032997
032998
032999
033000
033001
033002
033003
033004
033005
033006
033007
033008
033009
033010
033011
033012
033013
033014
033015
033016
033017
033018
033019
033020
033021
033022
033023
033024
033025
033026
033027
033028
033029
033030
033031
033032
033033
033034
033035
033036
033037
033038
033039
033040
033041
033042
033043
033044
033045

1.1 Technical Field

1.2 Related Art

The Internet has seen processing to specify and sometimes manipulate this information shift in location from being handled on the server providing a web site to the actual client computer (referred to herein as the client). One way of accomplishing scripting content for use on a client is through the use of Java.

1

Java applet 103 being received and run in browser 102. While Java applets contain text, they do not normally contain HTML content. Any text in an applet is generally displayed as in a window 104 separate from that of the display of the running Java applet 105. While most experienced users would not mind the opening and closing of various windows, novice users
5 may become flustered.

Further, Java applets are dynamically run on a user's machine. Because of the failure to store persistence information relating to previous interactions with the Java applet, navigation away from then back to the Java applet will re-execute the applet thereby replacing any previously entered information.

10 A further problem that needs to be overcome when forms are displayed as a result of activating an HTML hyperlink is that navigating away from the form then back to the form can be difficult. If the Java form was treated as HTML content is conventionally treated, then the form may be lost in a long history of visited pages. Further, with Java applets, only one form is generally opened at a time as only one applet is normally active. Navigating to a new
15 page or new form shuts down the previously running applet and fails to save previously entered information. Figure 2 shows a user initiating an applet 202 from browser 201. Prior to completing the interaction with applet 202, a user may choose to navigate to new page 203 or back to page 201. Attempting to re-navigate to applet 202 may completely refresh the running of the applet and destroy all previously entered information. For long forms, a user
20 will quickly tire of needing to reenter previously entered information

Accordingly, tighter integration between forms and content is needed.

2. Summary

A system and method for integrating forms and content is disclosed. By integrating both forms and content, the drawbacks of the prior art are reduced. The present invention
25 includes a container that houses an applet that displays both forms and HTML content. In

other words, the present invention displays static HTML data and dynamic Java form data in a single window. The invention permits a user to interact with a form created by a Java applet, navigate away, and have the form content maintained in the applet as the applet is not shut down. Also, as the invention maintains the content of the form with its incorporated HTML content, a user may open a new form without concern that the previous form and content will be deleted.

In one embodiment, the user may navigate static HTML hyperlinks to open new Java forms. The new form may be added to the running applet with a tab or other easy way of accessing the new form added to the graphical user interface displaying the forms. The user may navigate to HTML content from inside the running Java applet. Multiple forms may be opened simultaneously and may be accessed by tabs. In a further embodiment, an icon is displayed that, when activated, retrieves all forms and entered data. Activating the icon displays a graphical user interface that shows all open forms. The presence of this icon and its functionality assures a user that open forms and entered data may always be retrieved. By providing a way to access all open forms, the forms are not lost through a string of HTML navigations.

The invention also allows integration of HTML content and Java forms by enabling the user to open a new form by clicking an HTML hyperlink. New HTML content may also be displayed by navigating links in the displayed forms. These links or buttons may include Help buttons and the like. Activating these links or buttons redirects the applet to display the content of the HTML page associated with the button or link. In this regard, the applet is not shut down but rather the content of the new page is displayed in the applet housed by the container.

These and other aspects of the invention will be apparent from the following drawings and description.

3. Brief Description of Drawings

Figures 1A and 1B show conventional methods of displaying content and forms in browsers.

Figure 2 shows conventional navigation between forms and HTML pages.

5 Figure 3 shows the combination of forms and content in accordance with embodiments of the present invention.

Figure 4 shows the display of form content in response to user input in accordance with embodiments of the present invention.

Figure 5 shows an architecture for supporting the present invention.

10 Figure 6 shows a method for performing the present invention.

4. Detailed Description

The present invention relates to displaying both forms and content in a common graphical user interface. The present invention may be embodied as a Java applet that, when retrieved and run in a browser, the invention displays a form or forms to a user. The Java
15 applet is housed in a container. A container is an application program or subsystem in which the program building block known as a component is run. For example, a component - such as a button or other graphical user interface or a small calculator or database requestor - may be developed using JavaBeans that can run in Netscape containers such as browsers and in Microsoft containers such as Internet Explorer, Visual Basic®, and Word.

20 Figure 3 shows a form displayed in accordance with embodiments of the present invention. Form 302 is a form displayed in display 301. The contents of the form 302 are generated by a Java applet 304. The applet also contains HTML information 303. This information is also displayed in form 302. The Java applet may be downloaded from the Internet, retrieved locally, or obtained by any other means of retrieving information known in
25 the art. While Figure 1 displays the content of a Java applet apart from the content of browser

102 as window 104, the present invention combines both the HTML content and forms content together in a display of the form.

The browser display 301 includes back button 305 and icon (or button) 306. Activation of back button 305 directs the browser to display previously displayed content. As
5 is known in the art, as one navigates HTML pages, links to the pages are stored in a history stack (not shown for simplicity). By activating the back button 305, the browser is redirected to the top link in the history stack. The present invention includes the ability to add links to forms in the history stack.

As an example, the browser 301 displays pane 302 comprising a form. The form
10 includes HTML text 307 and buttons 311 and 312. The form also includes form fields 308 and 310. As shown in Figure 3, the form may be used as forms are normally used including in ordering books on-line, in e-commerce in general, in configuring systems (like a system gateway for a computer system), and other form related applications as are known in the art.

The system uses Java-based classes including JEditorPane to control when a form is
15 displayed. The form may include a variety of information as is known in the art. The system also instantiates objects from HTMLEditorKit classes to control the display of the HTML content 303 received from applet 304. Finally, the system uses HyperLinkListener to monitor for operation of hyperlinks in pane 302. While browser 301 may navigate to new content based on the activation of hyperlinks in it, operation of hyperlinks in pane 302 need to be
20 controlled so as to not refresh the content of any dynamic forms in pane 302. This is performed by having new forms or pages displayed in the running Java applet, rather than permitting the applet to be completely replaced by the browser's redirection to new content. With the use of the applets described here, one may use conventional browsers to realize the present invention.

In one embodiment, the present invention displays HTML content by instantiating `javax.swing.text.html.HTMLEditorKit` and `javax.swing.JEditorPane`. The `javax.swing.event.HyperlinkListener` interface is implemented by the classes in the invention. The various classes are described in additional detail in respect to Figure 5.

5 HyperEvents with the description field starting with "http://" cause the corresponding HTML page to be displayed in area 302. HyperlinkEvents with the description field starting with "form://" cause the container (specifically the workbench container 302 of Figure 4) to be displayed in the browser and the corresponding form to be added as a new tab in the container 302.

10 The invention may be practiced using Java Developer Kit 1.2.2-001 available from Sun Microsystems, Inc. This developer kit contains the Java Runtime Environment 1.2.2-001.

Figure 4 shows browser 301 having multiple active forms in pane 302. The forms are represented by tabs 401, 402, and 403. A new tab 405 is shown in broken lines. The contents of each tab are shown in display region 404 once the tab is selected. The content of each form, represented by tabs 401, 402, and 403, is separately maintained. Accordingly, navigation in window 404 does not refresh the currently displayed form or other forms (unless specified by, for example, navigation of a "reset" or "close" button as is known in the art). The Java forms get and set persistence information by communicating with a server process using one of the known communication protocols (for example, CORBA, RMI, or JDBC).

15

20

Figure 5 shows an architecture for supporting the invention. Applet 501 is from a class extending `javax.swing.JApplet`. Event manager 502 is a class implementing `java.awt.event.ActionListener`. History stack 503 is a stack class extending `java.util.Stack`. Workbench button 504 is a short cut button that, when activated, extends `javax.swing.JButton`. The action command field of this button is set to "wb://wb". Workbench

25

505 is a class extending javax.swing.JtabbedPane (resulting in the tabbed panes of Figure 4). Form factory 506 is a factory class as is known in the art that creates forms based on a string reference in the applet. Forms 507 and 508 are form classes extending javax.swing.JPanel.

HTML Panel 509 is a class for displaying HTML content extending
5 javax.swing.JEditorPane with a javax.swing.text.html.HTMLEditorKit object set as the current editor kit. This class also implements javax.swing.event.HyperlinkListener.

Back Button 510 is a navigation button extending javax.swing.JButton. The action or command field of this button is set to "back://back".

The flow of events are described as follows. All hyperlinks in the HTML content
10 displayed in HTML Panel 509 are handled by the Panel 509. When the Panel 509 receives a javax.swing.event.HyperlinkEvent, Panel 509 creates a java.awt.event.ActionEvent with the command field of the ActionEvent set to the description of the HyperlinkEvent. Panel 509 next forwards the ActionEvent to event manager 502. Workbench 505 listens to events from controls (or buttons operable by the user) on open form objects (507 and 508). The
15 workbench 505 may also perform some filtering by eliminating events that do not need additional action. All events that cause a new form or HTML page to become visible are forwarded to event manager 502. All other events are ignored. Finally, event manager 502 listens to events from buttons 504 and 510.

When a relevant event occurs, event manager 502 determines what action to take in
20 response to a java.awt.event.ActionEvent by examining the command field associated with the event. If the command field starts with "wb://", then the follow steps occur:

- 1) If the HTML panel 509 is visible, then it is hidden; and,
- 2) If the Workbench 505 is hidden, it is made visible.

If the command field starts with http: //, then the following steps are taken:

- 25
- 1) If the HTML panel 509 is hidden, then it is made visible;

- 2) If the Workbench 505 is visible, it is hidden;
- 3) The current page of the HTML Panel is set to the command field; and,
- 4) A new entry is placed in stack 503 with the contents of the command field.

If the command field starts with "form://" then the following steps are taken:

- 1) If the HTML panel 509 is visible, then it is hidden;
- 2) If the Workbench 505 is hidden, it is made visible;
- 3) The contents of the command field are passed to form factory 506 to create a new form object (for example, 507 and 508);
- 4) The newly created form object is added to the workbench 505; and
- 5) A new entry is placed in stack 503 with the contents set to "wb://wb".

If the command field starts with "back://", then the following steps are taken:

- 1) The top entry (or other entry) of stack 503 is removed (or popped as is known in the art); and,
- 2) The contents of the newly removed stack entry are used to set the command field of the current ActionCommand and the ActionCommand is reprocessed.

Figure 6 shows a method of creating and navigating form content in accordance with the present invention and graphically shows the addition of a new form and HTML. In step 601, a Java applet is run. In step 602, the applet instructs a browser to open a display window and populates the window with form information, in step 603. In step 604, the Java applet retrieves HTML content and forwards the content to the browser for display in the window.

In step 605, the Java applet running in the browser monitors navigation commands. These commands may take the form of events generated through selection of hyperlinks or other actions as are known in the art (including activation of buttons, icons, and the like). In step 606, the system determines if a new form is to be created. If so, the system adds a link to the currently displayed form or HTML page to the history stack (in step 607), displays the workbench container (in step 608) and adds a new tab (405) in the workbench container for accessing the new form (in step 609).

In step 610, the system determines if new HTML content is to be displayed. If so, the system adds a link to the current form or HTML page to the history stack (in step 611), hides the workbench container if visible (in step 612), and shows the new HTML page (in step 613).

Otherwise, the system continues to monitor for new actions or commands.

Various embodiments have been described. It is appreciated that various modifications of the embodiments are known to those of skill in the art and are considered within the scope of the present invention. For example, instead of using Java, one may use ActiveX controls to implement the current invention. Also, instead of using horizontally arranged tabs, one may use vertically arranged tabs. Further, one may substitute frames or a combination of frames and tabs to display forms.

The scope of the invention is intended to be limited only by the following claims.